LlamaDyno — Demo Walkthrough (send-to-anyone guide)

A self-contained guide for showing LlamaDyno to anyone — a teammate, a referee, a governing body, an investor — with **zero prior knowledge**. Follow it top to bottom and you'll go from a cold start to running a full, official speed test and exporting the record. No code, no install.

> **Who this is for.** Anyone giving or rehearsing a demo. You do **not** need to be a developer. Every button and field is named exactly as it appears on screen, in **bold**. For the day-to-day end-user manual see `07-runbook.md` (also in-app under **Help**, `/help`); for architecture see `PROJECT-STATE.md`.

> **Print-ready versions.** A print-ready PDF of this whole guide (`demo-walkthrough.pdf`) and a one-page **at-a-glance** slide (`demo-at-a-glance.pdf`) live alongside this file — hand those to non-technical folks. To film the explainer, use `demo-video-runboard.md`.

Contents

---

1. What LlamaDyno is (plain English)

Power-soccer is wheelchair soccer. To keep it fair, the sport caps how fast a powered chair may go. At a sanctioned event a chair is driven onto a **dyno** (a "rolling-road" — rollers that spin under the wheels and measure their speed), and a referee checks the chair against that cap **before** and **after** each game.

LlamaDyno is the **app that does the checking**. A referee drives the chair forward then reverse on the dyno; LlamaDyno reads the rollers live **in a web browser** (over a USB cable, using a browser feature called Web Serial), shows the speed, and judges each direction **PASS** or **FAIL** against the governing body's limit. It records an official result you can print or export, and it keeps working even if the venue Wi-Fi drops mid-event.

Who it's for: power-soccer **referees** and **officials** (run the tests), **club/region admins** (set up machines, rosters and rules), and **governing bodies** (define the speed limits). It replaces an old Windows-only desktop tool — LlamaDyno runs on any laptop or tablet, no install.

The one thing to remember for a demo: the headline feature is a **live, in-browser speed readout with a real PASS/FAIL verdict**. You can show that in 30 seconds with no account and no hardware (next section). Everything else — accounts, orgs, rosters, official records — is the scaffolding that turns one reading into a sanctioned record.

---

2. The 30-second no-setup self-demo

The fastest possible demo. No account, no hardware, works on the live site.

1. Open **https://llamadyno.app/guest?demo=1** in **desktop Chrome or Edge**.
2. It starts a **simulated chair** automatically. You'll see the status line **"Demo mode — simulated chair, no device connected."**
3. Watch the readout come alive: a big **averaged chair speed** number, **Left** and **Right** roller speeds, a **balance needle** (how evenly the two wheels drive), and a running **peak**. Because the simulated chair peaks around **6.8 mph** and the default limit is **6.2 mph**, the verdict flips to a red **FAIL** — which is exactly the moment that sells the product: *the app caught an over-speed chair.*
4. Tap **Classic** (top-right, under **View**) to switch the readout to the pixel-faithful legacy NEMI "Speed Check" dial, then **Enhanced** to switch back. Both are the same engine, two skins.
5. Want a PASS instead? Expand **Calibration constants**, raise **Maximum speed (mph)** to e.g. `8`, click **Apply** — the verdict recomputes to green **PASS**.

> **Why this works with no account:** `/guest` is the standalone reader (we call it Mode 1). It judges the live peak against the **Maximum speed** in its own calibration panel — no organization or ruleset needed — and **saves nothing**. It's the perfect "what does it do?" opener.

If you only have an iPad or a phone, skip to **§6 Lighter setups** — live USB reading needs desktop Chromium, but the demo still has a path.

---

3. Recommended demo setup (who holds what)

The richest demo shows the **full referee workflow with an official record**. You can do it solo (play both roles) or with a partner. You only need **one laptop**; a phone is a nice-to-have for the QR-scan beat.

Role	Device / browser	What they do	Where they are (URL)
You — Admin	Laptop, **Chrome/Edge**	Create the account & org, add a machine + ruleset + an athlete + a chair	**…**`/dashboard` → **Org console** (`/org`)
You — Referee	Same laptop	Make a tournament + match, run the guided capture, save & export	**Testing** (`/testing`)
(optional) Audience phone	Any phone camera	Scan a printed chair **QR label** during capture	n/a — points camera at the laptop/printout

> **Solo is fine.** A freshly created account is **both** the org admin **and** can run tests, so one person walks the whole story. The table above just names the two *hats* you'll switch between.

Before you present, have ready (5 min once): the account created, one **machine**, one **ruleset**, one **athlete**, one **chair**. The numbered steps below build all of them from scratch — do them once as a dry run, then you can re-run the *capture* beat (§4.6) live as many times as you like.

---

4. Full demo — cold start to official record

This is the complete path: **create account → set up the org → make it testable → the core capture → reset and re-run → export.** Do it on **desktop Chrome or Edge**.

Replace … with `https://llamadyno.app` (live) or `http://localhost:3000` (local).

4.1 Create your account (becomes the org admin)

1. Go to …/. On the landing page click **Create an organization** (or **Sign up** in the top bar).
2. On **"Create your account"** enter your **email** and a **password** (minimum **6 characters**), repeat it in **password confirmation**, and click **Create account**.
3. You're taken straight to **"Create your organization"**. Type an **Organization name** (e.g. `Bay Area Power Soccer`) and click **Create organization**. A slug is generated for you automatically.
4. You land on the dashboard with the banner **"<your org> is ready — you're its organization admin."** You are now a **tenant admin** — you can do everything below.

> **Why an org?** LlamaDyno is multi-tenant: every machine, chair, athlete, ruleset and result belongs to **one organization**. Creating the org makes you its admin and gives the referee flow something to scope to.

4.2 Open the Org console

In the top bar click **Org console** (or go to `/org`). The dashboard shows a **"Get set up"** checklist and counts. The left/top tabs are **Overview · Machines · Chairs · Teams · Athletes · Rulesets · Members · Settings**. You'll touch four of them.

4.3 Register a machine (the dyno + its calibration)

1. **Machines** → **Register machine**.
2. Enter a **Name** (e.g. `Dyno A`) and a **Serial**.
3. Under **Calibration**, the fields are **Left ConversionFactor (ft/count)**, **Right ConversionFactor (ft/count)**, **Left CountsPerRev**, **Right CountsPerRev**, **Minimum speed (mph)**, **Maximum speed (mph)**, **Speed tolerance (mph)**, and **Drum diameter (ft)**. For a demo the pre-filled placeholders are fine — just set **Maximum speed (mph)** to the ceiling you want to demo (e.g. `6.2`).
4. Click **Save**. You'll see **"Machine added."**

> The right roller's **CountsPerRev** is **negative** on purpose — the sign tells the reader which way the wheel turns, so forward motion reads positive on both wheels. Don't "fix" it. (Real calibration values are an operator input; placeholders ship until then.)

4.4 Choose a ruleset (the speed ceilings)

1. **Rulesets**. To start from a standard, find **"Clone a governing body"** and click **Clone →** next to one (e.g. USPSA / FIPFA) — that copies its default **Pre-game** and **Post-game** ceilings into an editable ruleset. Or click **New ruleset** to build one by hand.
2. Set the two ceilings under **Ceilings (mph) by phase** — **Pre-game** and **Post-game** — and click **Save**.

> **One ceiling per phase.** Pre-game and post-game each have their own limit, and **forward and reverse share** it. That's the whole adjudication rule.

4.5 Add one athlete and one chair

1. **Athletes → Add athlete**. Enter a **Name** (e.g. `Alex Rivera`) and an **Athlete number** (placard, e.g. `7`); **Team** is optional. **Save**. The athlete now shows as `Alex Rivera (#7)`.
2. **Chairs → Register chair**. Enter a **Serial number** (e.g. `BAPS-014`), a **Placard / number**, an **Ownership** (**Athlete owned** / **Program** / **Spare**) and optionally an **Associated athlete**. **Save** ("Chair registered.").
3. *(Optional, great on camera)* On the Chairs list click **Print QR labels → Print labels** to get a print-ready sheet of `LLAMADYNO:<serial>` QR codes, one per chair. Print or display one so you can **scan** it during capture.

You now have the minimum to produce a **real** PASS/FAIL: a **machine**, a **ruleset**, an **athlete** and a **chair**.

4.6 The core interaction — run a guided test

This is the beat to rehearse and re-run. Switch to your **referee** hat.

1. Top bar → **Testing** (or `/testing`) → **New tournament**. Enter a **Name**, a **Start date**, optionally a **Default ruleset**, then **Create tournament**.
2. On the tournament, click **New match**. Enter a **Name** (e.g. `Rockets vs Comets`); leave **Ruleset** blank to inherit the tournament's, or pick one. **Create match**.
3. On the match, click **New test**. The capture screen — **"Run a test"** — is deliberately uncluttered; click the **?** (top-right) to reveal inline tips beside each step. Now, top to bottom:
   - **Phase** — choose **Pre-game** or **Post-game** (this picks the ceiling).
   - **Dyno path** — **Out** or **In** (which way the chair drove on; flip it only if forward/reverse come out swapped).
   - **Machine** — pick `Dyno A`. Its calibration loads into the reader and gets stamped on the record.
   - In the reader, click **Try demo** (or **Connect device** with real hardware).
   - **Drive forward, pause ~2 s** → the **Forward reading** captures with a live **PASS/FAIL** badge. **Drive reverse, pause ~2 s** → the **Reverse reading** captures. (In demo mode the simulated chair does this for you.)
   - **Athlete** — pick `Alex Rivera (#7)` from the dropdown.
   - **Chair (scan QR)** — click **Scan QR** and point the camera at the label, pick from the list, or click **Simulate** to fill it for a walkthrough.
   - **Station** — optional (e.g. `Lane 2`), stamped on the record.
4. Click **Save test**. You return to the match with **"Test recorded — PASS."** (or **FAIL** / **recorded** if no ceiling applied), and the result appears in the table.

> **The 2-second pause is the secret handshake.** The app delimits one run from the next by ~2 seconds of stillness — that's how it knows the forward run ended and the reverse run is a separate reading.

4.7 Reset and re-run

- To re-test the **same** chair: in the reader click **Reset peak** — it clears both the forward and reverse readings and restarts the forward/reverse cadence. Drive again.
- To test the **next** athlete/chair: click **Save test**, then **New test** on the match page. Repeat §4.6 from step 3.
- The match page accumulates every result in a table (athlete, chair, phase, forward, reverse, verdict, referee, time).

4.8 Show the official record + export

On the match page, top-right:

- **Official record** → a clean, printable **"Official Speed-Test Record"** — org, tournament, match, ruleset, ceilings, every result with verdict and referee, a pass/fail summary, and signature lines. Click **Print / Save PDF** to produce the hardcopy.
- **Export JSON** → the stable machine-readable record (the same shape destined for PowerHub) for archiving or import.

That's the full story: **a referee caught an over-speed chair, recorded it, and the governing body has a signed, exportable record — all in a browser, all offline-safe.**

---

5. PowerHub mode — test live against PowerHub

Everything so far saved to **LlamaDyno's own database** (this is **Mode 2**). At a sanctioned USPSA event the system of record is **PowerHub** — the league's match-management app — and the dyno runs in **PowerHub mode (Mode 3)**: the chair reads *exactly* the same way, but the roster and the **pass/fail verdict come from PowerHub**, which adjudicates authoritatively. Nothing about the chair or its calibration changes — only **where the reading goes**.

5.1 Sign in with PowerHub credentials (no LlamaDyno account)

From the landing page tap **PowerHub**. Pick the **Environment** — **Development** to rehearse, **Production** for a real event (they're separate logins) — and sign in with the **referee's PowerHub email and password**. This is *not* a LlamaDyno login: PowerHub owns identity here, and the password is never stored.

5.2 Pick the match, pull the roster

You land on the **Match testing console** — PowerHub's live matches, each with its authoritative **criteria** read straight from PowerHub (e.g. **pre-game ≤ 6.2 mph, post-game ≤ 6.53 mph**, reverse required, one retest). Open a match to pull its roster.

5.3 Capture and push — PowerHub decides

Each dressed athlete is a card with their jersey number, their on-file readings, and their current eligibility. Drive the chair forward and reverse exactly as in the local flow, then tap **Send to PowerHub**. PowerHub re-adjudicates and returns the authoritative verdict — **eligible** or **off the game sheet** — which recolours the card; retest slots open automatically after a fail. (On **Development** only, a **Simulate** button can stand in for the dyno for a hands-free demo.)

---

6. Lighter setups

Pick the smallest demo that lands your point.

If you have…	Do this	Shows
30 seconds, any desktop	`…/guest?demo=1` (**§2**)	Live readout + PASS/FAIL, no account
A laptop, 5 min	§4.1–4.6 with **Try demo** instead of hardware	The whole referee workflow, no dyno

If you have…	Do this	Shows
A laptop + the real dyno	§4.6 with **Connect device** (pick the USB port)	The genuine end-to-end hardware path
Only an iPad / phone	Open `…/guest?demo=1` — the live USB reader is desktop-Chromium only, but **demo mode still animates** and you can talk through it	The product idea, without live USB
A PowerHub tournament	Landing → **PowerHub** → **Sign in to PowerHub** (no LlamaDyno account)	The live PowerHub overlay (**§5**)

---

7. FAQ — the real gotchas

Drawn from the actual code and on-screen messages — these are the things that trip a first-time demo.

Q. "Connect device" is greyed out / nothing happens. Web Serial is **Chromium-only**. On Safari, Firefox or iPad you'll see **"Web Serial isn't available here — use a desktop Chrome/Edge, or the Wi-Fi converter on iPad."** Use desktop **Chrome or Edge**, or fall back to **Try demo**.

Q. The demo shows FAIL — is something broken? No — that's intentional. The simulated chair peaks **~6.8 mph** against the default **6.2 mph** ceiling, so you can *see* the over-speed verdict. To show a PASS, raise **Maximum speed (mph)** in **Calibration constants** (guest screen) or on the **Machine** (referee flow).

Q. A reading won't capture. You must **pause ~2 seconds** after each run — that stillness is how the app separates the forward run from the reverse run. Use **Reset peak** to clear both and restart.

Q. Forward and reverse look swapped. Toggle **Dyno path** (**Out** / **In**) on the capture screen, or check which way the chair actually drove onto the road.

Q. The QR scan won't open the camera. The live scan needs the **BarcodeDetector** API (Chrome / Android). Where it's missing you'll see a note — pick the chair from the dropdown or click **Simulate** instead.

Q. The verdict says "—" instead of PASS/FAIL. No ceiling applied. On the guest screen, set **Maximum speed**. In the referee flow, make sure the **match has a ruleset** (its own or the tournament's default).

Q. I lost Wi-Fi mid-capture — did I lose the test? No. **Save test** while offline queues the result on the device (**"Saved offline — will sync when back online (N pending)."**) and the form resets for the next athlete. When you're back online it syncs automatically (**"Synced N queued results ✓"**) with **no duplicates**. Install the app (address-bar **Install** icon) for the most robust offline run. **Export JSON** is the manual fallback.

Q. I signed up but I'm not an admin / can't see the Org console. You become **tenant admin** of the org **you create** at sign-up. If you clicked **Skip for now**, create an org from the dashboard. The top-bar badge shows your active **org · role**.

Q. Which account is the super-admin? On deploy, the email in `ADMIN_EMAIL` (default `bpassmore@gmail.com`) is auto-promoted to **super admin** and lands on the **Master console**

(`/admin`). A normal demo account never needs this.

Q. I switched the readout to Classic and the colors don't follow dark mode. By design. **Classic** is a deliberately **theme-immune**, pixel-faithful copy of the legacy NEMI dial (locked red/green/black). **Enhanced** is the modern, theme-aware skin. Same numbers, different skin — toggle under **View**.

---

8. At a glance — addresses & commands

Live site: `https://llamadyno.app` · **Local:** `http://localhost:3000`

Want	Go to	Notes
No-setup self-demo	`/guest?demo=1`	Auto-starts simulated chair; desktop Chrome/Edge
Standalone reader	`/guest`	Live USB readout, nothing saved
Sign up / create org	`/users/sign_up` → `/organizations/new`	Landing **Create an organization**
Sign in	`/users/sign_in`	Password or **Sign in with a passkey**
Your dashboard	`/dashboard`	Org switcher + role home
Org admin setup	`/org`	Machines · Chairs · Teams · Athletes · Rulesets · Members · Settings
Print chair QR labels	`/org/chairs/labels`	**Print labels**
Referee workflow	`/testing`	Tournaments → Matches → **New test**
PowerHub overlay	landing **PowerHub** → `/power_hub/login`	PowerHub credentials, no LlamaDyno account
Master console	`/admin`	Super-admin only
In-app help	`/help`	Same material as `07-runbook.md`
Liveness check	`/healthz`	Public; liveness + baked theme only

Run it locally (developer machine):

```
bundle install
bin/rails db:prepare # create + migrate + seed (Postgres must be running)
bin/dev # boots Rails + Tailwind watch on :3000
```

Then open `http://localhost:3000/guest?demo=1`. Full developer setup, tests and deploy notes live in `PROJECT-STATE.md` and the `README`.

Every control, message and default in this guide is read from the live code (routes, controllers, views, the reader JS) as of the date of the commit that added it. If a label looks different on screen, the code changed — please update this file.